Hostile Intent Probability Engine

Millennial Software
&
UCCS Team:
Izzy Christopherson
Spencer Kardas
Kristen Poteat
Nick Ruvalcaba

TABLE OF CONTENTS

LIST OF FIGURES

1 EXECUTIVE SUMMARY

With the rapidly increasing number of foreign objects in orbit comes the rapidly increasing need for accurate observation, analysis and prediction. Foreign satellites with increasing capabilities absolutely have the potential to eliminate or disrupt government and/or defense operated assets. The issue lies within the lack of credible information and communication between foreign forces in the space domain. Without this information, it is imperative to observe, analyze, and diagnose the maneuvers of foreign satellites in order to prevent or to be able to take precautionary measures against hostile intentions.

Within that problem space lies a variety of necessary components; this project's scope only involves the analysis of the already observed and measured trajectories of target satellites. Before analysis can be done, the code will first propagate both satellites using an orbital propagation model to ensure consistent data across satellites. After this, the time and distance of the closest approach can be calculated and used in combination with capability data to accurately estimate the probability of hostile intent of the foreign satellite. Once probabilities for all satellite capabilities have been calculated, the code outputs an operator-friendly hostility score and all the data used obtained from it.

This system has been designed to operate under specific requirements that will allow it to be integrated into a larger program. These requirements involve latency, accuracy, and viability, and a degree of modularity and expandability, in case of future foreign satellite capability or further development. The problem space also demanded an objective and easily explainable analysis, which is why the final result was intentionally very overt.

With these requirements in mind, the chosen medium for calculation and data analysis was Python, receiving data in TLE format to dissect into indicators to which hostile probabilities could be extrapolated and output.

2 INTRODUCTION

Space Domain Awareness (SDA) is a critical component of modern space operations, enabling the monitoring and characterization of objects in Earth's orbit. As the number of active satellites and debris continues to increase, distinguishing between routine orbital behavior and potentially hostile actions has become an increasingly important challenge. Government and defense organizations require reliable tools to assess risks, protect assets, and support decision-making in an evolving space environment.

The objective of this project is to develop an algorithm capable of estimating the probability of hostile intent for an observed satellite using measured orbital data. The algorithm ingests time-tagged state information and derived orbital parameters to analyze satellite behavior. Using physics-based orbital models in combination with statistical inference methods, the system evaluates interactions between satellites and classifies behavior based on defined threat indicators.

The resulting output is repeatable and quantitative risk assessment that supports automated monitoring and defensive decision-making. By providing consistent and interpretable results, the system enhances situational awareness and improves the ability to identify potentially hostile satellite activity.

3 PROBLEM DESCRIPTION

Space Domain Awareness (SDA) systems must continuously monitor an increasingly congested orbital environment and distinguish between nominal satellite behavior and actions that may indicate hostile intent. The core challenge is not only detecting close approaches or anomalies, but determining whether those events represent credible threats based on physical feasibility, system capability, and contextual conditions.

The problem addressed in this project is the development of a system that can ingest heterogeneous orbital data sources and produce a quantitative, repeatable assessment of potential hostile intent between space objects. Specifically, the system must evaluate whether an observed (red) satellite poses a threat to a reference (blue) satellite by analyzing relative motion, encounter geometry, and the feasibility of multiple interaction types.

This problem involves several key technical challenges:

- **Data Integration and Standardization:** Orbital data may be provided in different formats (e.g., TLEs, state vectors) and reference frames. The system must convert these inputs into a consistent internal representation suitable for analysis.
- **Time Synchronization and Propagation:** Satellite states are defined at different epochs and must be propagated to a common time horizon to enable meaningful comparison.
- **Encounter Detection:** The system must efficiently identify potential conjunction events by screening large time windows for proximity conditions.
- **Uncertainty Modeling:** Satellite position and velocity are inherently uncertain, requiring covariance modeling and propagation to accurately characterize encounter conditions.
- **Behavior Feasibility Assessment:** Hostile actions such as collision, jamming, grappling, and directed energy engagement must be evaluated based on both geometry (distance, timing) and known system capabilities.
- **Probabilistic Risk Estimation:** Rather than binary classification, the system must compute likelihoods of each behavior using statistical methods such as Monte Carlo sampling.

The system must also satisfy several systems engineering requirements, including accuracy, low latency, scalability, and modularity. It must produce explainable outputs that allow operators to understand how each probability was derived and to adjust parameters such as thresholds and capability constraints as needed.

The final objective is to generate a structured output consisting of:

- Time of Closest Approach (TCA)
- Distance of Closest Approach (DCA)
- Behavior-specific probabilities (collision, jamming, grappling, directed energy)
- Supporting metadata for traceability and decision-making

By framing the problem as a modular, probabilistic assessment pipeline, the system enables consistent evaluation of satellite interactions and supports automated threat analysis within SDA workflows.

4 CONCEPTUAL DESIGN

Two main approaches were considered for the Hostile Intent Assessment System to evaluate how close another satellite gets and whether that behavior should be considered hostile. Both concepts were based on using the relative position of a secondary satellite compared to a primary satellite and then using that distance to classify behavior. Since our entire system is software based and works with incoming data, both concepts were able to meet most of the functional and non-functional requirements without much difficulty. Because of that, the focus of this evaluation was not whether the concepts could meet requirements, but rather how they compared in terms of simplicity, accuracy, and ease of implementation.

4.1 Evaluation Criteria

4.1.1 Requirement Based Criteria

Both concepts were checked against the main system requirements, including:

- Automatically ingesting and processing satellite event data
- Determining potential hostile intent based on position and behavior
- Producing a clear output indicating whether something is considered hostile
- Logging data and results for traceability
- Maintaining a modular structure that could be expanded later

Since both designs rely on basic distance calculations and standard data handling, they were both capable of meeting these requirements.

4.1.2 Performance-Based Metrics

To better compare the concepts, we also looked at how the system performed using test data. The following metrics were used:

- Data ingestion latency
- Classification latency
- State vector comparison accuracy
- Event characterization precision
- Anomaly detection recall
- Intent inference precision
- Contextual enrichment coverage

These were generated using our validation script and helped confirm that the selected design behaves as expected.

4.1.3 Performance Quality Attributes

We also considered overall system qualities such as:

- How easy the results are to understand
- How simple the system is to implement
- Efficiency and scalability

- Ability to adjust or tune parameters
- Modularity for future improvements

4.2 Candidate Concepts

4.2.1 Concept A – Multiregional Cube Based Detection

This concept uses multiple cubes centered around the primary satellite, which is treated as a point at the center. Each cube represents a different distance range and depending on which region a secondary satellite falls into, the system would assign a level of hostility.

Strengths:

- Allows for multiple levels of classification instead of just yes/no
- Gives a more structured way to break up space into regions

Limitations:

- More complicated to implement and keep track of multiple regions
- Distances are not consistent in all directions such as corners of a cube are farther than faces
- Harder to interpret results compared to a simpler model

4.2.2 Concept B – Spherical Radius Based Detection

This concept uses a single sphere centered around the primary satellite, which is treated as a point. If another satellite falls within the radius of the sphere, it is classified as hostile. The radius can be adjusted depending on the scenario being evaluated.

Strengths:

- Same distance in every direction
- Much simpler to calculate using radial distance
- Easy to understand output
- Easier to maintain and radius can be adjusted depending on the situation

Limitations:

- Only gives a binary result instead of multiple levels
- Less detailed than the cube-based approach

4.3 Concept Screening and Feasibility Assessment

4.3.1 Feasibility

Both concepts were feasible to implement within the scope of this project. They rely on basic geometry and standard programming tools, so no additional hardware or complex external systems were required.

4.3.2 Technology Readiness

Both approaches use well known methods such as distance calculations and coordinate comparisons, so there were no concerns about needing new or untested technology.

4.3.3 Requirement Fit

Since both concepts are based on comparing satellite positions, they both meet the main system requirements. The main difference between them is how they interpret distance and how complex the classification logic becomes.

4.4 Concept Comparison

Even though both concepts meet the requirements, we compared them to see which one would work better overall.

Criteria	**Cube-Based Concept**	**Sphere-Based Concept**
Requirement Satisfaction	Met	Met
Computational Complexity	Higher	Lower
Geometric Accuracy	Lower	Higher
Output Interpretability	Moderate	High
Implementation Simplicity	Lower	Higher
Scalability	Similar	Similar

One of the biggest differences is how distance is managed. A cube does not represent equal distance in all directions, which can introduce inconsistencies. A sphere, on the other hand, is the same distance from the center in every direction, which makes it a better representation of proximity.

4.5 Final Concept Selection and Justification

The spherical radius-based approach was selected for the final design. The main reason for selecting the spherical model was its simplicity and consistency. Because the distance from the center is the same in every direction, it provides a more accurate way to determine how close another satellite is. It also requires less logic and is easier to implement compared to managing multiple cube regions.

The binary output also makes the results easier to understand, which is important for operator use. Even though it does not provide multiple levels of classification, the adjustable radius still allows flexibility depending on the scenario.

4.6 Modeling Assumptions

In both concepts, the primary satellite was treated as a point located at the center of the coordinate system. This simplifies the calculations and avoids having to model the actual shape of the satellite, which is negligible compared to the distances being analyzed.

4.7 Conclusion

Based on the evaluation, the spherical detection method was selected as the best option for this project. It provides a good balance between accuracy, simplicity, and performance, while still meeting the system requirements. Although further testing with real data is needed, the current design provides a solid starting point and can be expanded or improved in future work.

5 TESTING AND ANALYSIS SUMMARY

The final design was tested in order to adequately prove its ability to meet the expectations and requirements set at the beginning of the project design. These requirements are accuracy and precision of calculations and propagations, classification and data ingestion latency, and hostility probability and contextualization.

In order to test the system, validation cases and measurement functions were coded to run alongside the hostile probability main function. Using generated testing data, each requirement was tested, and the results of the test can be found in the table below. Each engineering parameter set in the design phase of the project was met by the system and validated by the measurement functions within.

Requirement	Target	Verification Method	Outcome
Data Ingestion Latency	≤ 5 seconds per update	Timing tests using simulated orbital data ingestion runs	Met (≤ 5 seconds)
State Vector Comparison Accuracy	≥ 95% agreement across sources	Cross-source comparison against truth/reference data	Met (≥ 95%)
Kafka Messaging Throughput	Dropped	N/A	N/A
Event Characterization Precision	≥ 90% accuracy vs. truth data	Monte Carlo scenarios compared to truth data	Met (≥ 90%)
Anomaly Detection Recall	≥ 85% on validated test cases	Validated anomaly test cases	Met (≥ 85%)
Intent Inference Precision	≥ 80% validated accuracy	Behavioral probability assessment scenarios	Met (≥ 80%)
Classification Latency	≤ 3 seconds per event	Runtime performance timing tests	Met (≤ 3 seconds)

Indicator Weighting Adaptability	Operator-adjustable (0–1 scale)	Parameter sensitivity/ configuration/ testing	Met (fully adjustable)
Contextual Enrichment Coverage	≥ 90% of cataloged satellites enriched	Metadata enrichment tests against representative catalog	Met (≥ 90%)
Modularity / Extensibility	Add new source/indicator in <1 day, no refactor	Modularity demonstration via indicator addition	Met (< 1 day integration, no refactor)

FINAL DESIGN DESCRIPTION

The final design consists of a modular algorithm developed to evaluate the probability of hostile intent between satellites using time-tagged orbital data and physics-based modeling. The system integrates data ingestion, orbital propagation, encounter analysis and behavior-specific probability modules to generate a quantitative and repeatable assessment of potential threats.

5.1 Structure

A high-level operation view of the system is shown in Figure 1. This diagram illustrates the overall decision-making process, beginning with the ingestion of orbital data for both the observed (red) and reference (blue) satellites. The system evaluates whether the observed satellite enters a region of interest relative to the reference satellite and determines if further analysis is required. If conditions are met, the system proceeds to evaluate potential hostile behaviors.

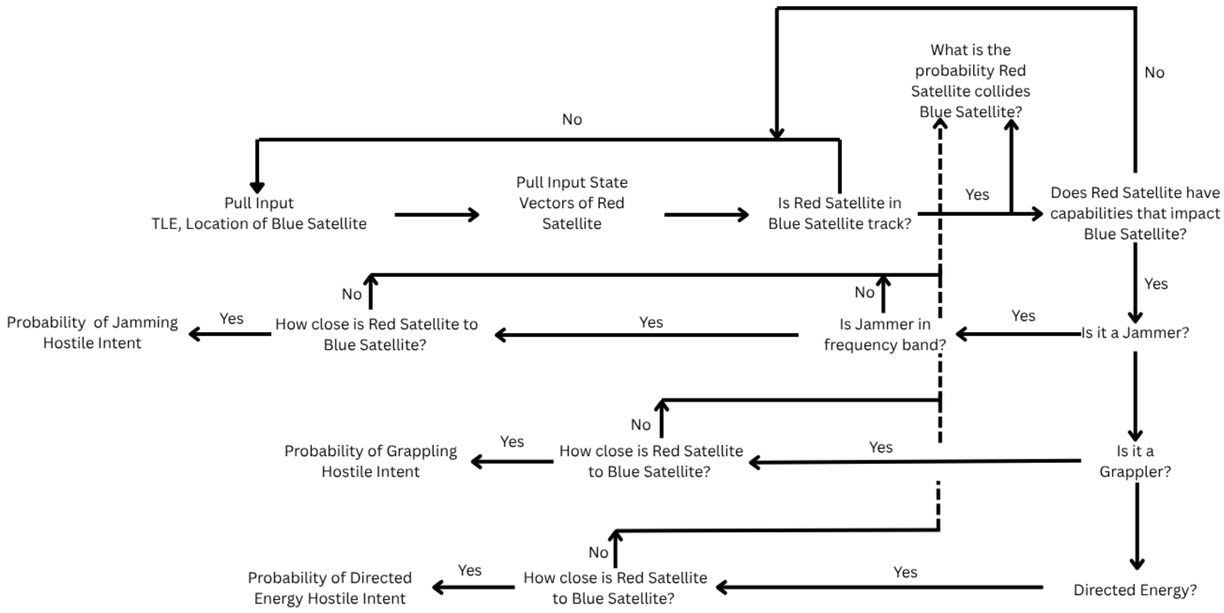

Figure 1. Operational Overview of Hostile Intent Assessment System (OV-1)

The system begins with data ingestion, where Two-Line Element (TLE) data and state vectors are collected for both satellites. These inputs are standardized to ensure consistency in reference frames, units, and time formats. The system then propagates both satellite states to a common time horizon using a physics-based orbital propagation model. This step ensures that all subsequent calculations are performed using synchronized state information.

Following propagation, the system performs encounter screening to determine whether a potential interaction exists between the satellites. If an encounter is identified, key parameters are computed, including the time of closest approach (TCA) and the distance of closest approach (DCA). These values define the relative geometry of the interaction and serve as the primary inputs for further analysis. In addition, uncertainty in satellite position is modeled using covariance propagation, allowing the system to account for variations in predicted position at the time of closest approach.

5.2 Operational Views

The detailed functional workflows for each behavior-specific module are shown in Figures 2 through 5. Each module follows a consistent structure consisting of propagation, encounter screening, parameter computation, and probability evaluation. This standardized approach ensures consistency across all behavior assessments while allowing for behavior-specific constraints to be applied.

Figure 2 shows the collision probability module. In this module, the system evaluates whether the satellites are on intersecting trajectories based on proximity and relative motion. The probability of collision is determined using TCA, DCA, and encounter covariance, which define both the likelihood and uncertainty of a close approach.

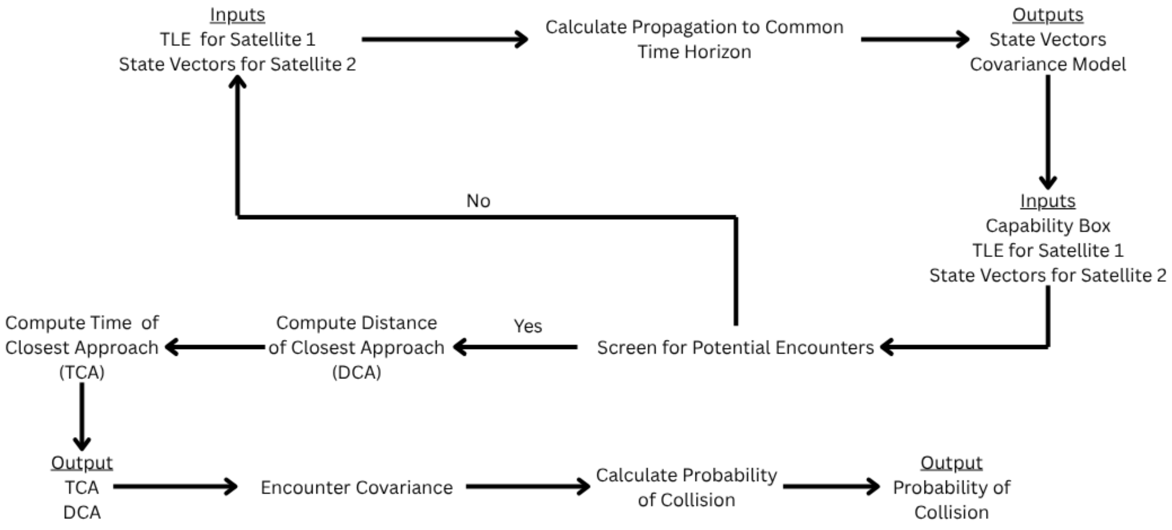

Figure 2: Collision Probability Module Workflow

Figure 3 shows the jamming assessment module. In addition to proximity, this module evaluates whether the observed satellite possesses jamming capabilities and whether it operates within a relevant frequency band. If these conditions are satisfied, the system calculates the probability of jamming based on distance, alignment and capability constraints.

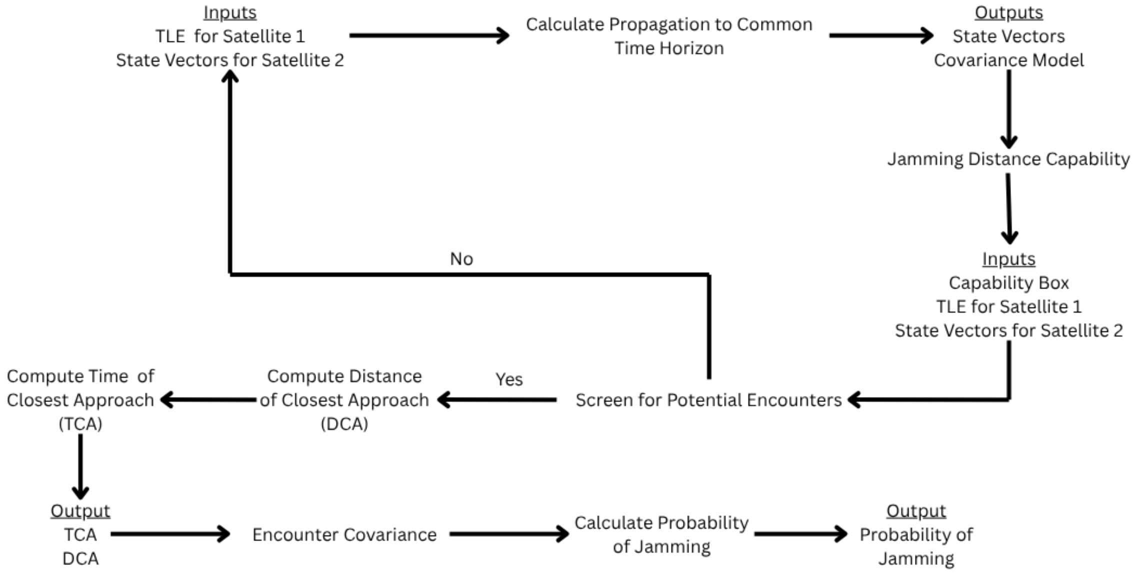

Figure 3: Jamming Probability Module Workflow

Figure 4 shows the grappling module, which evaluates the likelihood of physical interaction between satellites. This assessment focuses on close-range proximity and relative positioning, determining whether the observed satellite is capable of approaching and maintaining conditions required for grappling.

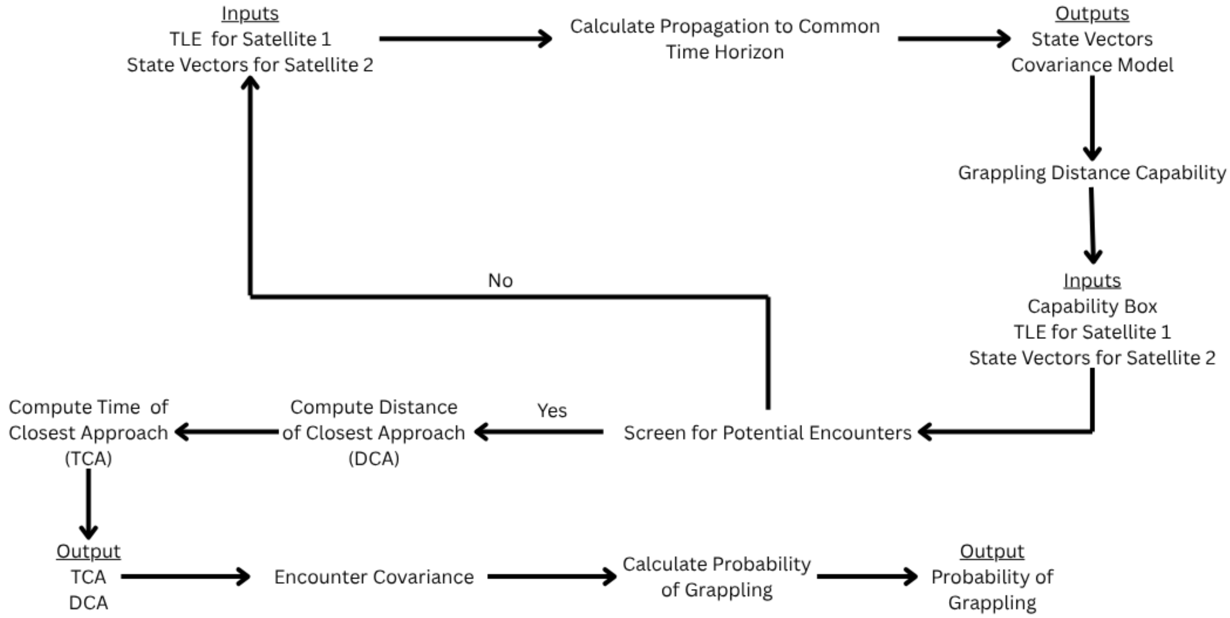

Figure 4: Grappling Probability Module Workflow

Figure 5 shows the directed energy module. This module evaluates whether the observed satellite has the capability to apply directed energy and whether it is within an effective operational range. The probability is determined based on distance and capability constraints.

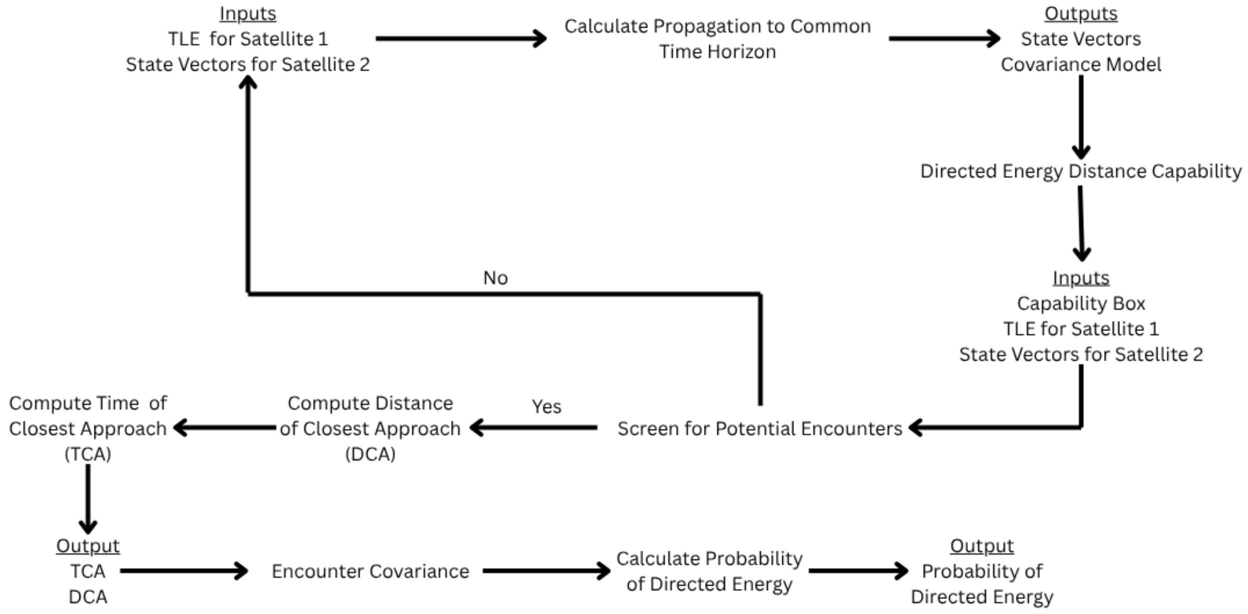

Figure 5: Directed Energy Probability Module Workflow

Each module produces a probability value representing the likelihood of the corresponding hostile behavior. These outputs are derived from encounter geometry, capability constraints, and uncertainty modeling. The individual probabilities are then combined to generate an overall hostile intent assessment, along with intermediate outputs that provide transparency and traceability of the results.

The system inputs and outputs are formally defined using input/output specification sheets. Inputs include satellite orbital data such as TLE sets, state vectors and associated metadata. Outputs include probability values for each behavior, as well as supporting parameters: TCA, DCA, and encounter covariance. These definitions ensure consistency, repeatability and validation of the system. Detailed input/output definitions are provided in Appendix A. These include formal specifications for all system inputs, outputs, and other variables throughout the algorithm.

5.3 Pseudocode

The hostile intent assessment system is implemented as a structured, modular pipeline that evaluates whether a Red object poses a threat to a Blue satellite. The algorithm follows a sequence of preprocessing, geometric analysis, probabilistic modeling, and decision-making steps to produce a final hostility score and classification.

At a high level, the system first standardizes and validates all input data, ensuring consistency in coordinate frames, units, and time systems. It then applies a **track-relevance gate**, which determines whether the Red object is close enough to the Blue satellite to warrant further analysis. If the object is not relevant, the system terminates early and produces a low-risk assessment.

If the track gate passes, the system evaluates multiple potential threat capabilities, including collision, jamming, grappling, and directed energy. Each capability is analyzed independently using orbital propagation, encounter detection, and probabilistic modeling. These results are then combined using configurable weights to produce a final hostility score.

In addition to geometric and probabilistic analysis, the system incorporates **contextual factors and historical comparisons**. Each assessment is stored, compared against previous results, and used to generate intent indicators such as repeated close approaches, maneuver changes, and increasing hostility trends. This allows the system to move beyond static geometry and incorporate behavioral reasoning.

The final output includes a numerical hostility score, a classification label, and structured reasoning that explains how the assessment was derived.

5.4 Function

From an operational perspective, the system is executed by providing the required orbital data inputs, running the algorithm over a defined time horizon, and analyzing the resulting probability outputs. Operators can use both the overall hostility assessment and individual behavior probabilities to support monitoring and decision-making.

The final design emphasizes modularity, scalability, and transparency. Each behavior module can be independently modified or expanded without affecting the overall system architecture. By combining physics-based modeling with probabilistic analysis, the system provides a robust and explainable framework for evaluating hostile intent in space domain awareness applications.

6 PROJECT, SPONSOR & TEAM REFLECTION

6.1 Project Planning & Project Management

The project planning process initially followed a structured systems engineering approach, including the development of a project plan and Gantt chart to define key milestones and deliverables. These planning tools were used to organize development phases such as algorithm design, implementation, validation, and visualization.

However, throughout the project, the team encountered challenges related to evolving or cutting requirements from the project sponsor. The sponsor’s expectations and desired features changed multiple times during development, which required the team to frequently revise the project plan and adjust the Gantt chart accordingly. This introduced uncertainty in scheduling and made it difficult to maintain a consistent development timeline.

Additionally, communication delays from the sponsor resulted in periods where progress was slowed due to a lack of clear direction or confirmation of requirements. As a result, the team had to make assumptions to continue development, which occasionally led to rework when updated guidance was later provided.

Despite these challenges, the team adapted by maintaining flexibility in the project plan and prioritizing a modular system design. This allowed components of the system to be updated or modified without requiring complete redesigns. Through this process, the team gained valuable experience in managing uncertainty, adapting to changing requirements, and maintaining progress in a dynamic project environment.

6.2 Sponsor Interactions

Communication with the project sponsor was an important part of the development process, but it also presented several challenges. Interactions with the sponsor occurred through meetings and email communication; however, the frequency of communication was relatively low. This made it difficult to obtain timely feedback on design decisions and implementation progress.

In addition, the sponsor’s requirements and expectations changed throughout the duration of the project. While evolving requirements are common in real-world systems engineering projects, the combination of slightly infrequent communication and changing expectations created uncertainty for the team. At times, this resulted in revisiting previously completed work or adjusting system functionality to align with updated requirements.

To address these challenges, the team focused on maintaining forward progress by implementing a flexible and modular system design. This approach allowed updates to be incorporated without significantly impacting the overall system structure. The team also documented assumptions and design decisions to make it easier to adapt when new information became available.

Overall, this experience highlighted the importance of consistent communication with stakeholders and reinforced the need for adaptability when working with evolving sponsor requirements.

6.3 Team Interactions

Communication within the team played a critical role in the successful completion of the project. The team maintained regular communication through both informal discussions and scheduled check-ins, which helped ensure that all members remained aligned on goals, progress, and responsibilities.

Due to the challenges associated with sponsor communication and changing requirements, strong internal communication became especially important. Team members collaborated frequently to interpret requirements, make design decisions, and adjust the implementation as needed. This collaborative approach allowed the team to continue progressing even when external guidance was limited.

There were also occasional challenges related to expectations and contributions within the team. At times, not all members were contributing equally or meeting the expected level of involvement. However, these issues were addressed through direct communication within the team. By discussing expectations and responsibilities openly, the team was able to resolve these concerns and improve overall participation and accountability.

The team was effective in dividing responsibilities across different areas of the project, such as core algorithm development, validation, and visualization, while still maintaining awareness of the overall system. This balance between individual contributions and team coordination supported successful integration of all components.

One area for improvement would be maintaining more detailed internal documentation of evolving requirements to reduce potential rework. However, overall, the team maintained a productive working dynamic and demonstrated strong adaptability and problem-solving throughout the project.

This experience reinforced the importance of clear communication, accountability, and teamwork when working on complex engineering projects with changing requirements.

7 CONCLUSION

The final result of this project is a python system with the ability to rapidly and accurately determine the probability of hostile intent of a foreign satellite through propagation and orbital modeling when given the measured information of a pair of satellites. In addition, the project can determine the probabilities of multiple capabilities, which are collision, grappling, directed energy, and jamming. This system is designed to integrate within a larger project and meets all the requirements relative to that proposed system.

8 FINAL DESIGN PACKAGE

9 GLOSSARY

Epoch	The specific reference time at which orbital data (such as a TLE or state vector) is defined and valid.
State Vector	A mathematical representation of a satellite's orbital state at a given epoch, consisting of position (r) and velocity (v) vectors in a defined reference frame.
Position Vector (r)	A three-dimensional vector defining a satellite's location relative to the center of the Earth at a specific epoch.
Velocity Vector (v)	A three-dimensional vector defining a satellite's speed and direction of motion at a specific epoch.
Two-Line Element (TLE)	A standardized orbital data format consisting of two text lines that describe a satellite's orbit at a specific epoch.
Propagator	A mathematical model used to predict a satellite's future position and velocity by advancing its orbital state forward or backward in time from an epoch.
Propagation	The process of computing a satellite's position and velocity at future or past times using a propagator.
Common Time Horizon	A shared time interval over which multiple satellites are propagated so their relative positions can be directly compared.
Ephemeris	A time-ordered set of satellite positions and velocities generated through orbit propagation.
Relative Position	The vector difference between two satellites' position vectors at the same time, used to compute separation distance and encounter geometry.
Time of Closest Approach (TCA)	The time at which the distance between two satellites is minimized during an encounter.
Distance of Closest Approach (DCA)	The minimum separation distance between two satellites during an encounter.
Covariance	A mathematical representation of uncertainty in a satellite's state vector, describing the expected spread and correlation of position and velocity errors.
Encounter Covariance	The combined uncertainty of two satellites' relative positions at the time of closest approach, typically expressed in an encounter-specific reference frame.
Gaussian Assumption	The assumption that position and velocity uncertainties follow a normal (Gaussian) probability distribution.
Directed Energy Capability	The set of system constraints (such as range, geometry, and dwell time) that define whether a directed energy system can effectively engage a target.
Probability of Directed Energy	A computed likelihood that directed energy engagement criteria are satisfied during an encounter, based on geometry, uncertainty, and system capability.
Hard-Body Radius	The effective physical radius of a satellite used to model collision geometry, typically representing the maximum extent of the object.
Combined Collision Radius	The sum of the hard-body radii of two satellites, defining the distance threshold at which a physical collision is assumed to occur.

Probability of Collision	A computed likelihood that two satellites will physically collide during an encounter, based on relative geometry and state uncertainty.
Collision Region	The spatial region defined by the combined collision radius within a physical collision is considered to occur
Encounter Geometry	The relative spatial relationship between two satellites during a close approach, typically characterized by TCA and DCA.
Conjunction	An event in which two satellites pass within a defined proximity of each other, potentially resulting in a collision.
Screening Threshold	A predefined distance used to identify candidate conjunctions requiring higher-fidelity collision analysis.
Relative Covariance	The covariance describing uncertainty in the relative position of two satellites, derived from individual satellite covariances.
Encounter Frame	A reference frame defined at the time of closest approach, used to express relative position and covariance for probability calculations.
Linearization Assumption	The assumption that relative satellite motion can be approximated as linear in the vicinity of the time of closest approach.
Grappling	A proximity operation in which one satellite attempts to physically capture or attach to another satellite.
Grappling Distance Capability	The maximum relative distance and geometric envelope within which a grappling system can successfully capture a target satellite.
Capture Envelope	The three-dimensional spatial region around a target satellite within which grappling is considered feasible.
Probability of Grappling	A computed likelihood that the relative position uncertainty of two satellites places the target within the grappling capture envelope during an encounter.
Grappling Feasibility	A determination of whether grappling conditions are met based on relative geometry, distance, and system capability constraints.
Jamming	An interference action intended to disrupt, degrade, or deny a satellite’s communications or sensing capability without physical contact.
Jamming Distance Capability	The maximum relative distance within which a jamming system can effectively interfere with a target satellite.
Probability of Jamming	A computed likelihood that jamming conditions are satisfied during a satellite encounter, based on relative geometry and state uncertainty.
Jamming Feasibility	A determination of whether relative distance and geometry satisfy the constraints required for effective jamming.
Jamming Constraint	A defined spatial limit or threshold used to assess whether jamming can occur during an encounter.
Line-of-Sight Assumption	The assumption that no physical obstructions block signal propagation between satellites during a jamming attempt.
Spatial Proximity Constraint	A distance-based requirement that must be met for jamming to be considered feasible.

10 APPENDIX A: INPUT/OUTPUT DEFINITIONS

10.1 Directed Energy

	Title	Purpose	Input	Output
Box 1	TLE for Satellite 1, State Vectors for Satellite 2	Ingest Initial orbital data for both objects	Sat 1: TLE, Sat 2: State vector at epoch (r,v) + epoch time	Parsed orbit representations for both satellites at their respective epochs, Metadata: epoch times, reference frames, units
Box 2	Calculate Propagation to Common Time Horizon	Propagate both satellites to a shared time window so they can be compared consistently.	Orbits/epochs from Box 1, Common horizon definition (start time, end time, step size), Propagator choice	Time-aligned ephemerides: r(t), v(t) for Sat 1 and Sat 2 over the horizon
Box 3	State Vectors and Covariance Model	Attach uncertainty to the propagated states using a covariance model.	Propagated states from Box 2, Covariance at epoch for each object, Covariance propagation method	Covariance vs time for each satellite
Box 4	Directed Energy Distance Capability	Determine whether a directed energy system can "reach" based on distance/geometry constraints	System capability parameters (max effective range, min range, dwell constraints), Relative geometry (distance vs time from Box 2)	A function/limit
Box 5	Capability Box, TLE for Satellite 1, State Vectors for Satellite 2	Combine orbital data and capability constraints into a single assessment dataset.	Ephemerides (Box 2), Covariance (Box 3), Capability rules (Box 4)	Prepared dataset for screening: relative position/time series + capability thresholds

Box 6	Screen for Potential Encounters	Quickly filter the time horizon to find candidate close-approach intervals.	Relative distance vs time (from Box 2/5), Screening thresholds (distance < X km)	Candidate encounter time brackets to refine with exact DCA/TCA.
Box 7	Compute Distance of Closest Approach	For each candidate window, compute the minimum separation distance.	Candidate encounter window (Box 6), Relative motion data	DCA (minimum) per encounter window
Box 8	Compute Time of Closest Approach	Determine the time at which the minimum separation occurs	Candidate encounter window (Box 6), Relative motion data	TCA per encounter window
Box 9	TCA DCA	Package the encounter geometry result as the key pair.	TCA (Box 8), DCA (Box 7)	Encounter summary record: TCA,DCA
Box 10	Encounter Covariance	Compute covariance at encounter time	Covariance models (Box 3), TCA (Box 9) Frame definition for encounter analysis	Relative covariance at TCA
Box 11	Calculate Probability of Directed Energy	Use geometry + uncertainty + capability to compute probability that directed energy conditions are met.	TCA/DCA (Box 9). Encounter covariance (Box 10), Capability constraints (Box 4/5), Engagement criteria	Probability value per encounter
Box 12	Probability of Directed Energy	Final reported metric/output product	Probability (Box 11)	Final probability report (number, metadata, time, assumptions)

10.2 Grappling

	Title	Purpose	Input	Output
Box 1	TLE for Satellite 1, State Vectors for Satellite 2	Ingest Initial orbital data for both objects	Sat 1: TLE, Sat 2: State vector at epoch (r,v) + epoch time	Parsed orbit representations for both satellites at their respective epochs, Metadata: epoch times, reference frames, units
Box 2	Calculate Propagation to Common Time Horizon	Propagate both satellites to a shared time window so they can be compared consistently.	Orbits/epochs from Box 1, Common horizon definition (start tome, end time, step size), Propagator choice	Time-aligned ephemerides: r(t), v(t) for Sat 1 and Sat 2 over the horizon
Box 3	State Vectors and Covariance Model	Attach uncertainty to the propagated states using a covariance model.	Propagated states from Box 2, Covariance at epoch for each object, Covariance propagation method	Covariance vs time for each satellite
Box 4	Grappling Distance Capability	Determine whether grappling is feasible based on required proximity and any basic geometric constraints.	Grappling capability parameters(maximum grappling distance/capture envelope size, allowable approach geometry constraints if applicable), Relative geometry (distance vs time from Box 2)	Grappling feasibility constraint/function (distance threshold model)
Box 5	Capability Box, TLE for Satellite 1, State Vectors for Satellite 2	Combine orbital data, uncertainty information, and grappling capability constraints into a single dataset for encounter screening	Ephemerides from Box 2, Covariance information from Box 3, Grappling distance capability constraints from Box 4	Prepared dataset for screening, including relative position time history and grappling distance thresholds.
Box 6	Screen for Potential Encounters	Quickly filter the time horizon to find candidate close-approach intervals.	Relative distance vs time (from Box 2/5), Screening thresholds (distance < X km)	Candidate encounter time brackets to refine with exact DCA/TCA.
Box 7	Compute Distance of Closest Approach	For each candidate window, compute the minimum separation distance.	Candidate encounter window (Box 6), Relative motion data	DCA (minimum) per encounter window
Box 8	Compute Time of Closest Approach	Determine the time at which the	Candidate encounter window (Box 6), Relative motion data	TCA per encounter window

		minimum separation occurs		
Box 9	TCA/DCA	Package the encounter geometry result as the key pair.	TCA (Box 8), DCA (Box 7)	Encounter summary record: {TCA,DCA}
Box 10	Calculate probability of grappling	Use encounter geometry and uncertainty to compute the probability that the target is within the grappling capture envelope at closest approach	TCA/DCA (Box 9), Encounter covariance (Box 9), Grappling capability constraints (Box 4/5), Capture envelope definition (effective grappling radius/volume)	Probability value per encounter
Box 11	Probability of Grappling	Final reported metric product for grappling feasibility assessment.	Probability of grappling (Box 10)	Final probability report (number and metadata such as encounter time window, TCA/DCA, and modeling assumptions)

10.3 Jamming

	Title	Purpose	Input	Output
Box 1	TLE for Satellite 1, State Vectors for Satellite 2	Ingest Initial orbital data for both objects	Sat 1: TLE, Sat 2: State vector at epoch (r,v) + epoch time	Parsed orbit representations for both satellites at their respective epochs, Metadata: epoch times, reference frames, units
Box 2	Calculate Propagation to Common Time Horizon	Propagate both satellites to a shared time window so they can be compared consistently.	Orbits/epochs from Box 1, Common horizon definition (start tome, end time, step size), Propagator choice	Time-aligned ephemerides: r(t), v(t) for Sat 1 and Sat 2 over the horizon
Box 3	State Vectors and Covariance Model	Attach uncertainty to the propagated states using a covariance model.	Propagated states from Box 2, Covariance at epoch for each object, Covariance propagation method	Covariance vs time for each satellite
Box 4	Jamming Distance Capability	Determine whether a jamming system can effectively interfere with a target satellite based on relative distance constrains.	Jamming system capability parameters (maximum effective jamming range, minimum rang if applicable), relative geometry	Jamming distance constraint or feasibility function
Box 5	Capability Box	Combine orbital data, uncertainty information, and jamming capability constraints into a single dataset for encounter screening.	Ephemerides from Box 2, Covariance information from Box 3, Jamming distance capability constraints from Box 4	Prepared dataset for screening, including relative position time history and jamming distance thresholds.
Box 6	Screen for Potential Encounters	Quickly filter the time horizon to find candidate close-approach intervals.	Relative distance vs time (from Box 2/5), Screening thresholds (distance < X km)	Candidate encounter time brackets to refine with exact DCA/TCA.
Box 7	Compute Distance of Closest Approach	For each candidate window, compute the minimum separation distance.	Candidate encounter window (Box 6), Relative motion data	DCA (minimum) per encounter window
Box 8	Compute Time of Closest Approach	Determine the time at which the minimum separation occurs	Candidate encounter window (Box 6), Relative motion data	TCA per encounter window

Box 9	TCA DCA	Package the encounter geometry result as the key pair.	TCA (Box 8), DCA (Box 7)	Encounter summary record: {TCA,DCA}
Box 10	Encounter Covariance	Compute covariance at encounter time	Covariance models (Box 3), TCA (Box 9) Frame definition for encounter analysis	Relative covariance at TCA
Box 11	Calculate Probability of Jamming	Compute the likelihood that jamming conditions are satisfied during a satellite encounter.	TCA/DCA (Box 9), Encounter covariance (Box 10), Jamming distance capability constraints from Box4/5	Probability of jamming per encounter
Box 12	Probability of Jamming	Provide the final jamming feasibility metric for reporting or decision-making	Probability of jamming (Box 11)	Final probability of jamming

10.4 Collision

	Title	Purpose	Input	Output
Box 1	TLE for Satellite 1, State Vectors for Satellite 2	Ingest Initial orbital data for both objects	Sat 1: TLE, Sat 2: State vector at epoch (r,v) + epoch time	Parsed orbit representations for both satellites at their respective epochs, Metadata: epoch times, reference frames, units
Box 2	Calculate Propagation to Common Time Horizon	Propagate both satellites to a shared time window so they can be compared consistently.	Orbits/epochs from Box 1, Common horizon definition (start tome, end time, step size), Propagator choice	Time-aligned ephemerides: r(t), v(t) for Sat 1 and Sat 2 over the horizon
Box 3	State Vectors and Covariance Model	Attach uncertainty to the propagated states using a covariance model.	Propagated states from Box 2, Covariance at epoch for each object, Covariance propagation method	Covariance vs time for each satellite
Box 4	Collision Geometry Parameters	Define collision-related geometric parameters used to assess physical overlap.	Hard-body radius Satellite 1, Hard-body radius Satellite 2	Combines collision radius
Box 5	Collision Assessment Inputs	Combine orbital data and collision geometry parameters into a single assessment dataset.	Ephemerides from Box 2, Covariance from Box 3, Collision geometry parameters from Box 4.	Prepared dataset for collision screening and analysis
Box 6	Screen for Potential Encounters	Quickly filter the time horizon to find candidate close-approach intervals.	Relative distance vs time (from Box 2/5), Screening thresholds (distance < X km)	Candidate encounter time brackets to refine with exact DCA/TCA.
Box 7	Compute Distance of Closest Approach	For each candidate window, compute the minimum separation distance.	Candidate encounter window (Box 6), Relative motion data	DCA (minimum) per encounter window
Box 8	Compute Time of Closest Approach	Determine the time at which the minimum separation occurs	Candidate encounter window (Box 6), Relative motion data	TCA per encounter window
Box 9	TCA DCA	Package the encounter geometry result as the key pair.	TCA (Box 8), DCA (Box 7)	Encounter summary record: {TCA,DCA}

Box 10	Calculate probability of collision	Estimate the likelihood that the satellites collide during the encounter.	TCA/DCA (Box 9), Encounter covariance (Box 3/Box 9), Combined collision radius (Box 4)	Probability of collision per encounter
Box 11	Probability of Collision	Provide the final collision risk metric.	Collision probability from Box 10	Final probability of collision

11 APPENDIX B: ALGORITHMS

11.1 Orbit Propagation Algorithm (SGP4)

SGP4 is an analytical propagation algorithm used to convert TLE data into position and velocity vectors. Rather than numerically integrating gravitational forces, SGP4 applies perturbation theory to evolve mean orbital elements forward in time, accounting for Earth oblateness and atmospheric drag through simplified models. Due to TLEs being generated using this same dynamic model, SGP4 reproduces the satellite trajectory.

Perturbation theory: Approximates real orbital motion by applying small analytical corrections to an ideal two-body orbit.

Physical model: SGP4 takes the TLE and reconstructs how the orbit moves over time by applying simplified physics.

Accounts for:

- Earth gravity (not spherical)
- Atmospheric drag
- Periodic oscillations

Computes:

$$x(t) = f(mean\ elements\ at\ epoch, \Delta t)$$

f is a long analytical perturbation solution. Extremely fast and stable.

Output:

For any requested time

$$SGP4(TLE, t) \rightarrow \begin{bmatrix} r(t) \\ v(t) \end{bmatrix}$$

Position and velocity in an Earth-centered inertial frame.

Purpose: Propagate two satellites defined by TLEs from their individual epochs to a shared analysis time grid so their positions and velocities can be directly compared at identical times.

- Inputs
    - sat1_TLE_line1, sat1_TLE_line2
    - sat2_TLE_line1, sat2_TLE_line2
    - analysis_start_time
    - analysis_end_time
    - time_step [seconds]
- Outputs
    - analysis_times
    - sat1_position, sat1_velocity
    - sat2_position, sat2_velocity

- Assumptions
    - SGP4 is the correct propagator for TLE inputs
    - Atmospheric drag is implicitly modeled via the BSTAR coefficient
    - Step size is sufficiently small to capture closest approach
    - No maneuvering occurs during the analysis window

Procedure

1. Construct common analysis time grid

$$\Delta t = t - t_{epoch}$$

$$t_k = t_{start} + k\Delta t_{grid}, k = 0,1, \dots, k$$

2. Initialize SGP4 satellite objects
3. Propagate Satellite 1 to each analysis time
4. Propagate Satellite 2 to each analysis time
5. Return synchronized ephemerides

For each satellite and time (t_k):

$$x(t_k) = \begin{bmatrix} r(t_k) \\ v(t_k) \end{bmatrix}$$

$$x(t) = SGP4(mean\ elements, BSTAR, \Delta t)$$

Each TLE is propagated using the SGP4 analytical orbit model to a shared discrete UTC time grid, producing synchronized TEME-frame position and velocity vectors for conjunction assessment

11.2 Covariance Propagation Algorithm

Purpose: Propagate each satellite's 6x6 state covariance from its epoch to the common analysis time grid, consistent with the SGP4 state propagation, then form the relative covariance for probabilistic assessment.

- Inputs
  For each satellite:
    - tle_line1, tle_line2
    - initial_epoch_time (from TLE)
    - initial_state_covariance_P0 (6x6)
    - analysis_times (common grid from orbit propagation box)
- Outputs:
  For each satellite:
    - covariance_over_time_P[k] (6x6 at each analysis time)
    - relative_covariance_Prel[k] = P1[k] + P2[k]
- Assumptions:
    - State errors are approximately Gaussian
    - Satellite errors are independent
    - SGP4 outputs are in TEME

Procedure

1. Build Initial Covariance

$$\sigma_x = \sigma_y = \sigma_z = 1\ km$$

$$\sigma_{\dot{x}} = \sigma_{\dot{y}} = \sigma_{\dot{z}} = 1 * 10^{-5}\ km/s$$

2. Build Diagonal Covariance in 6x6 matrix

$$P_0 = \begin{bmatrix} \sigma_x^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{\dot{z}}^2 \end{bmatrix}$$

3. Build Matrix F

$$r = [x, y, z]$$

$$r_{mag} = \sqrt{x^2 + y^2 + z^2}$$

4. Compute matrix G:

$$G = -\mu * \left(\frac{I}{r_{mag}^3} - 3 * \frac{r * r^T}{r_{mag}^5} \right)$$

   I = identity 3x3

5. Build the dynamics matrix A:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ G11 & G12 & G13 & 0 & 0 & 0 \\ G21 & G22 & G23 & 0 & 0 & 0 \\ G31 & G32 & G33 & 0 & 0 & 0 \end{bmatrix}$$

6. Convert A into step matrix F:

$$F = I + A * dt$$

   I = 6x6 identity matrix

   dt = small timestep

7. Update Covariance:

$$P_{new} = F * P_{old} * F^T$$

8. Do for each satellite:
   Satellite 1 uncertainty – P1(t)

   Satellite 2 uncertainty – P2(t)

9. Combine Uncertainties:
   Relative Covariance:

(uncertainty of the miss distance)

$$P_{rel} = P1 + P2$$

11.3 Monte Carlo Probabilistic Assessment

- Inputs
  - Mean states from SGP4 at the Time of Closest Approach
    $$\hat{x}_1 = \hat{x}_1(TCA), \hat{x}_2 = \hat{x}_2(TCA)$$
  - Covariances at the same time:
    - P1=P1(TCA)(6x6), P2=P2(TCA)(6x6)
    - $P_{rel}(TCA) = P1(TCA) + P2(TCA)$
  - Combined (hard-body) collision radius:
    $$R_c = R_1 + R_2$$
  - N = number of Monte Carlo Samples
- Outputs:
  - Estimated probability

Procedure

1. Compute relative mean state:

$$\hat{x}_{rel}(TCA) = \hat{x}_1(TCA) - \hat{x}_2(TCA)$$

2. Split into relative position and velocity:

$$\hat{x}_{rel} = \begin{bmatrix} \hat{r}_{rel} \\ \hat{v}_{rel} \end{bmatrix}, \hat{r}_{rel} = \hat{r}_1 - \hat{r}_2$$

best estimate of the miss vector

3. Constructing the Sampling Matrix L

L is a lower triangle matrix

Use matrix P for this

Diagonal Elements:

$$l_{ii} = \sqrt{p_{ii} - \sum_{k=1}^{i-1} l_{ik}^2}$$

Off-Diagonal Elements:

$$l_{ij} = \frac{p_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}}{l_{jj}}$$

4. Convert covariance into a sampling matrix

$$P_{rel} = LL^T$$

5. Generate Monte Carlo samples of relative state

i = 1 to N

6. Draw a random vector

$$z \sim N(0, I_6)$$

$$z = [z1, z2, z3, z4, z5, z6]$$

7. Convert random numbers into a physical error

$$\delta x = [dx, dy, dz, dvx, dvy, dvz]$$

$$\delta x = L * z$$

8. Create a possible true state

$$x_{rel}^{(i)} = \widehat{x_{rel}} + \delta x$$

9. Extract position

$$r_{rel}^{(i)} = [x, y, z]$$

10. Compute the separation distance

$$d^{(i)} = \sqrt{x^2 + y^2 + z^2}$$

11. Nominal Distance Definition:

$$DCA = ||\hat{r}_{rel}(TCA)||$$

DCA is the nominal miss distance about which the Monte Carlo samples are distributed using the relative covariance.

11.4 Probability Conditions

Probability is evaluated at the closest-approach event because this produces the maximum likelihood of interaction.

Collision condition: $d^{(i)} \leq R_c$
Compare with collision radius

$$R_c = R_1 + R_2$$

If true, then satellites overlap in space

Directed Energy condition:$R_{min} \leq d^{(i)} \leq R_{max}$
Directed Energy Engagement Event: $E_{DE}^{(i)} = \begin{cases} 1, if\ all\ DE\ conditions\ are\ satisfied \\ 0, otherwise \end{cases}$
Probability Estimate
Count successes: $N_{success} = \sum_{i=1}^{N} E_{DE}^{(i)}$
Compute probability: $P_{DE} = \frac{N_{success}}{N}$

Grappling condition: $d^{(i)} \leq R_g$
R_g=Grappler arm length

Grappling event: $E_g^{(i)} = \begin{cases} 1, if\ capture\ conditions\ are\ satisfied \\ 0, otherwise \end{cases}$

Probability Estimate

Count successes: $N_{success} = \sum_{i=1}^{N} E_g^{(i)}$

Compute probability: $P_g = \frac{N_{success}}{N}$

Jamming Condition: $R_{min} \leq d^{(i)} \leq R_j$

R_j=maximum effective jamming distance

R_{min}=minimum distance

Jamming Event: $E_{jam}^{(i)} = \begin{cases} 1, if\ jamming\ conditions\ are\ satisfied \\ 0, otherwise \end{cases}$

Count successes: $N_{success} = \sum_{i=1}^{N} E_{jam}^{(i)}$

Compute probability: $P_{jam} = \frac{N_{success}}{N}$

12 SOURCES

[1] Analytical Graphics, Inc. (AGI), "STK 11.3 Documentation – Vehicle: Satellite Orbit Propagation Messages." [Online]. Available: https://help.agi.com/stk/11.3/index.htm#stk/vehSat_orbitProp_msgp4.htm. [Accessed: Feb. 20, 2026].

[2] NASA Office of the Chief Engineer (OCE), "NASA OCE-51," NASA Headquarters, Washington, DC, USA. [Online]. Available: https://nodis3.gsfc.nasa.gov/OCE_docs/OCE_51.pdf. [Accessed: Feb. 20, 2026].

[3] J. M. Hanson and B. B. Beard, "Applying Monte Carlo Simulation to Launch Vehicle Design and Requirements Verification," presented at the AIAA Conference, 2010. [Online]. Available: https://ntrs.nasa.gov/api/citations/20100035661/downloads/20100035661.pdf.

[4] C. York, D. Geller, and D. Woffinden, "Generalized Augmented-State Covariance Analysis for Spaceflight," in Proc. 47th AAS Guidance, Navigation and Control (GN&C) Conf., Breckenridge, CO, USA, Jan. 31–Feb. 5, 2025. [Online]. Available: https://ntrs.nasa.gov/api/citations/20250000396/downloads/GeneralizedAugmentedStateCov2.pdf.

[5] GeeksforGeeks, "What is Monte Carlo Simulation?," Jul. 17, 2025. [Online]. Available: https://www.geeksforgeeks.org/artificial-intelligence/what-is-monte-carlo-simulation/.